Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A method comprising:

allocating a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, with a portion of the memory entry ~~includes~~ including a unique identifier assigned to the instructions.

2. (Original) The method of claim 1, further comprising:

maintaining a count of threads included in the multithreaded engine that use the memory entry.

3. (Original) The method of claim 1, further comprising:

maintaining a bit to represent availability of the memory entry for thread use.

4. (Original) The method of claim 2 wherein maintaining the count includes incrementing the count to represent a thread initiating use of the memory entry.

5. (Original) The method of claim 2 wherein maintaining the count includes decrementing the count to represent a thread halting use of the memory entry.

6. (Original) The method of claim 3 wherein maintaining the bit includes setting the bit to represent availability of the memory entry for thread use.

7. (Original) The method of claim 3 wherein maintaining the bit includes clearing the bit to represent unavailability of the memory entry for thread use.

8. (Original) The method of claim 3, further comprising:

checking the bit to determine the availability of the memory entry for thread use.

9. (Original) The method of claim 1 wherein the unique identifier includes four bits.

10. (Original) The method of claim 1 wherein the memory entry identifies a location in a local memory included in the multithreaded engine of the packet processor.

11. (Currently amended) A computer program product, tangibly embodied in a machine-readable medium, the computer program product being operable to cause a machine to:

allocate a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, with a portion of the memory entry ~~includes~~ including a unique identifier assigned to the instructions.

12. (Original) The computer program product of claim 11 being further operable to cause a machine to:

maintain a count of threads included in the multithreaded engine that use the memory entry.

13. (Original) The computer program product of claim 11 being further operable to cause a machine to:

maintain a bit to represent availability of the memory entry for thread use.

14. (Original) The computer program product of claim 12 wherein maintaining the count includes incrementing the count to represent a thread initiating use of the memory entry.

15. (Original) The computer program product of claim 12 wherein maintaining the count includes decrementing the count to represent a thread halting use of the memory entry.

16. (Original) The computer program product of claim 13 wherein maintaining the bit includes setting the bit to represent availability of the memory entry for thread use.

17. (Original) The computer program product of claim 13 wherein maintaining the bit includes clearing the bit to represent unavailability of the memory entry for thread use.

18. (Original) The computer program product of claim 13 being further operable to cause a machine to:

check the bit to determine the availability of the memory entry for thread use.

19. (Original) The computer program product of claim 11 wherein the unique identifier includes four bits.

20. (Original) The computer program product of claim 11 wherein the memory entry identifies a location in a local memory included in the multithreaded engine of the packet processor.

21. (Currently amended) A memory manager comprises:

a process to allocate a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, with a portion of the memory entry ~~includes~~ including a unique identifier assigned to the instructions.

22. (Original) The memory manager of claim 21, further comprises:

a process to maintain a count of threads included in the multithreaded engine that use the memory entry.

23. (Original) The memory manager of claim 21, further comprises:

a process to maintain a bit to represent availability of the memory entry for thread use.

24. (Currently amended) A system comprising:

a packet processor to allocate a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, with a portion of the memory entry ~~includes~~ including a unique identifier assigned to the instructions.

25. (Previously presented) The system of claim 24 wherein the packet processor is further configured to:

maintain a count of threads included in the multithreaded engine that use the memory entry.

26. (Previously presented) The system of claim 24 wherein the packet processor is further configured to:

maintain a bit to represent availability of the memory entry for thread use.

27. (Currently amended) A network forwarding device comprising:

an input port for receiving packets;

an output for delivering the received packets; and

a network processor to allocate a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, with a portion of the memory entry ~~includes~~ including a unique identifier assigned to the instructions.

28. (Previously presented) The network forwarding device of claim 27, wherein the network processor is further configured to maintain a count of threads included in the multithreaded engine that use the memory entry.

29. (Previously presented) The network forwarding device of claim 28, wherein the network processor is further configured to maintain a bit to represent availability of the memory entry for thread use.

30. (Currently amended) A method comprising:

allocating a content-addressable-memory (CAM) entry to a microblock executed on a multithreaded microengine included in a network processor, with a portion of the CAM entry ~~includes~~ including a unique identifier assigned to the microblock.

31. (Original) The method of claim 30, further comprising:

maintaining a count of threads included in the multithreaded microengine that use the CAM entry.

32. (Original) The method of claim 30, further comprising:

maintaining a bit in a status register to represent availability of the CAM entry to identify a local memory location.

33. (Previously presented) The method of claim 1, wherein the memory entry comprises a content-addressable memory entry.

34. (Previously presented) The computer program product of claim 11, wherein the memory entry comprises a content-addressable memory entry.

35. (Previously presented) The memory manager of claim 21, wherein the memory entry comprises a content-addressable memory entry.

36. (Previously presented) The system of claim 24, wherein the memory entry comprises a content-addressable memory entry.

37. (Currently amended) The network forwarding device of claim 27, wherein the memory entry comprises a content-addressable memory entry.